

# ALGORITMOS EVOLUTIVOS Y SU APLICABILIDAD EN LA TAREA DE CLASIFICACIÓN

A. Villagra, D. Pandolfi, M. Lasso, M. de San Pedro  
Laboratorio de Tecnologías Emergentes (LabTEM)  
Proyecto UNPA-29/B064<sup>1</sup>  
División Tecnología  
Unidad Académica Caleta Olivia  
Universidad Nacional de La Patagonia Austral  
Ruta 3 Acceso Norte s/n  
(9011) Caleta Olivia – Santa Cruz - Argentina  
e-mail: { avillagra, dpandolfi, mlasso, edesanpedro }@uaco.unpa.edu.ar  
Tel/Fax (0297) 4854888 Int 144

G. Leguizamón  
Laboratorio de Investigación y Desarrollo en Inteligencia Computacional (LIDIC)  
Departamento de Informática  
Universidad Nacional de San Luis  
Ejército de los Andes 950 - Local 106  
(5700) - San Luis -Argentina  
e-mail: legui@unsl.edu.ar  
Tel:(02652) 420823 / Fax:(02652) 430224

## Resumen

En este artículo se describe en forma breve una de las líneas de investigación que se están llevando a cabo en el Laboratorio de Tecnologías Emergentes (LabTEM) sobre Algoritmos Evolutivos y su aplicabilidad como técnica alternativa y/o complementario en tareas de Minería de Datos, específicamente clasificación.

Factores como el avance tecnológico asociado al continuo abaratamiento de los costos, hace que los volúmenes de datos almacenados crezca exponencialmente. En la actualidad, estamos en una etapa en la que no es fácil visualizar e interpretar los datos que están almacenados. Existen muchos dominios en los cuales la acumulación de datos es altísima y por consiguiente se hace cada vez más difícil poder obtener información relevante para la toma de decisiones basadas en dichos datos.

La tarea de Minería de Datos implica "escabar" en esa inmensidad de datos, en búsqueda de patrones, asociaciones o predicciones que permitan transformar esa maraña de datos en información útil. En particular esta línea de investigación aplicará Algoritmos Evolutivos en la de la tarea de clasificación.

## 1. Introducción

En los últimos años, ha existido un gran crecimiento en nuestras capacidades de generar y coleccionar datos, debido básicamente al gran poder de procesamiento de las máquinas y a su bajo costo de almacenamiento. Sin embargo, dentro de estas enormes masas de datos existe una gran cantidad de información "oculta", de gran importancia estratégica, a la que no se puede acceder por las técnicas clásicas de recuperación de la información.

En [1] se define la minería de datos como el proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos. Esta tarea no

siempre parte de un conocimiento previo de lo que se busca en el conjunto de datos disponibles, por el contrario, es muy frecuente que no sepamos de antemano lo que se busca. Es decir, se realiza una búsqueda de patrones desconociendo aquel que pueda surgir.

La minería de datos tiene dos grandes retos: por un lado, trabajar con grandes volúmenes de datos, procedentes principalmente de sistemas de información, con los problemas que esto conlleva (ruido, datos ausentes, intratabilidad, volatilidad de los datos, entre otras cosas) y por otro lado, usar técnicas adecuadas para analizar los mismos y extraer conocimiento novedoso y útil. La minería de datos consiste principalmente de dos pasos, pre-procesamiento de datos, durante el cual las características o atributos relevantes se extraen de los datos, y reconocimiento en estas características de patrones no triviales de información útil.

El pre-procesamiento de los datos es una tarea crítica y consume mucho tiempo. Para asegurar el éxito del proceso de minería de datos, es importante que las características extraídas sean relevantes al problema y representativas de los datos. Dependiendo del tipo de datos que esta siendo minado, la actividad de pre-procesamiento puede requerir varias sub-actividades. Si la cantidad de datos es muy grande, se puede tomar una muestra y realizar el trabajo con menos instancias. Se elimina el ruido en los datos y se extraen las características relevantes. A veces los datos provienen de diversas fuentes y se requiere una fusión de los mismos para que puedan ser minados. Al finalizar esta actividad, se tiene un vector de características para cada instancia de datos. Dependiendo del problema y de los datos, puede ser necesario reducir el número de características usando selección de características o técnicas de reducción de la dimensión. Luego que esta actividad de pre-procesamiento ha finalizado, los datos están preparados para detectar los patrones través del uso de algoritmos que llevan a cabo alguna de las siguientes tareas: clasificación, clustering, regresión, etc.

<sup>1</sup> El Grupo de Investigación cuenta con el apoyo de la Universidad Nacional de La Patagonia Austral.

Por lo anteriormente expuesto es importante diferenciar en la minería de datos, el tipo de tareas que se suelen abordar y las técnicas utilizadas en cada caso. Como ejemplos de tareas generales se pueden mencionar el aprendizaje de conceptos, clasificación, categorización, regresión, agrupamiento (o clustering), correlaciones y análisis de asociación. Estas tareas pueden ser abordadas mediante distintos métodos o técnicas que suelen adaptarse mejor de acuerdo a la tarea sobre la cual se trabajará. Entre las técnicas más conocidas se puede mencionar el aprendizaje de reglas de clasificación, reglas de asociación, reglas relacionales, reglas difusas, árboles de decisión (y regresión), ecuaciones de regresión, redes neuronales, metaheurísticas, etc.

Los Algoritmos Evolutivos (AEs) son metaheurísticas que emplean modelos computacionales del proceso evolutivo. Existen una gran variedad de AEs, los principales incluyen: Algoritmos Genéticos [2,3], Programación Evolutiva [4,5], Estrategias Evolutivas [6,7] y Programación Genética [8]. Todos estos algoritmos comparten un concepto base común que es simular la evolución de los individuos que forman la población usando un conjunto de operadores predefinidos. Comúnmente se usan dos tipos de operadores: de selección y de búsqueda. Los operadores de búsqueda más usados son la mutación y la recombinación.

El operador de selección depende principalmente de la medida percibida de la aptitud (fitness) de cada individuo y da fuerza a la selección natural y a la supervivencia del que más se ajusta. Los operadores de recombinación y mutación perturban en forma estocástica a los individuos proveyendo eficiencia en la exploración del espacio de búsqueda. Esta perturbación se controla principalmente por el usuario que define el porcentaje de recombinación y el de mutación. Aunque simplista del punto de vista de un biólogo, estos algoritmos son lo suficientemente complejos para proporcionar mecanismos de búsqueda robustos y poderosos y han mostrado su fuerza resolviendo duros problemas de optimización.

Los AEs son una de las metaheurísticas más difundidas en la actualidad. Los mismos están siendo usados para resolver una gran variedad de problemas del mundo real, entre estos problemas, se encuentra el desafío de la minería de datos.

En relación al uso de AEs en minería de datos, en [9,10], Cantú Paz et al. [11] y [12] se describen algunas de las aplicaciones más importantes de esta técnica en las distintas etapas de la extracción de conocimiento. Particularizando en actividades recientes trabajos como [13,14] utilizan a los algoritmos evolutivos en la selección de características, en clasificación [15,16,17,18,19,20] y en clustering [21,22, 23].

## **2. Algoritmos Evolutivos en la tarea de clasificación**

En AEs hay dos maneras principales de representar conjuntos de reglas. En el enfoque de "Michigan" [24, 25], cada individuo en la población representa una regla de longitud fija, y la población entera representa el objetivo. En contraste, en el enfoque de "Pittsburgh" [26,27,28], cada individuo de tamaño variable representa un conjunto entero de reglas. Las dos representaciones tienen sus méritos e inconvenientes y se han usado con éxito en sistemas clasificadores que son

sistemas basados en reglas que combinan refuerzo de aprendizaje y algoritmos evolutivos.

Existen trabajos en los que se discute el uso de algoritmos evolutivos, particularmente, algoritmos genéticos y la programación genética, en minería de datos y descubrimiento de conocimiento enfocándose en la tarea de clasificación [10]. En particular, se estudia cómo la representación de los individuos, los operadores genéticos y la función de aptitud tienen que ser adaptados para extraer conocimiento de alto nivel de los datos.

Otros enfoques se basan en programación genética para generar prototipos en un problema de clasificación [15]. En este contexto, el conjunto de prototipos con los cuales se puede determinar el origen de las muestras de un conjunto de datos, se codifica con un multiárbol, es decir un conjunto de árboles, que representan el cromosoma. A diferencia de otros enfoques, los cromosomas son de longitud variable y esto permite tratar aquellos problemas de clasificación en los cuales una o más clases tienen subclases.

Se ha desarrollado una estructura para un clasificador basado en reglas difusas genéticas. Primero, el problema de clasificación se divide en varios problemas de dos clases siguiendo el esquema de binarización de clases difusa. Luego se evoluciona la regla difusa para las dos clases de problemas usando el enfoque de aprendizaje interactivo Michigan, y finalmente las reglas evolucionadas se integran usando el esquema de binarización de clases difuso [16].

Los algoritmos genéticos y las redes neuronales artificiales (ANNs) se han usado juntas para entrenar o ayudar en el entrenamiento de las redes, para buscar los pesos de la red, para buscar los parámetros de aprendizaje apropiados, para reducir el tamaño del conjunto de entrenamiento seleccionando las características más pertinentes o para diseñar la estructura de la red [29,30,31]. Entrenar una ANN es una tarea de optimización con la meta de encontrar un conjunto de pesos que minimizan alguna medida de error. El espacio de búsqueda tiene muchas dimensiones y es probable que contenga óptimos locales múltiples. Los AEs evitan quedar atrapados en un óptimo local probando simultáneamente las regiones múltiples del espacio. Algunos algoritmos comienzan con una población aleatoria y usan los pesos encontrados por el AE en la red sin ningún refinamiento [32, 33, 34]. Otros usan backpropagation u otros métodos para refinar los pesos encontrados por el AE [35,36]. La motivación para este enfoque es que los AEs rápidamente identifican regiones prometedoras del espacio de búsqueda, pero no adecuan los parámetros muy rápidamente. Así que, se usa AEs para encontrar un prometedor conjunto inicial de pesos desde el cual un método basado en el gradiente puede alcanzar rápidamente un óptimo. En otros casos los algoritmos refinan los resultados encontrados por un algoritmo de aprendizaje NN. Aunque los AEs no refinan las soluciones muy rápido, ha habido algunos intentos de mejorar la población inicial del AE con soluciones encontradas con backpropagation [37].

Además, ciertos enfoques usan AEs para diseñar la topología de una ANN: usando una codificación directa para especificar cada conexión de la red o evolucionando una especificación indirecta de la conectividad.

La idea importante detrás de las codificaciones directas es que una red neuronal puede considerarse como un grafo dirigido donde cada nodo representa una neurona y cada arco es una conexión. Un método común de

representar grafos dirigidos es una matriz binaria de conectividad. La matriz de conectividad puede ser fácilmente representada en el AE y varios trabajos han usado este enfoque con éxito [38, 39]. Usando este método se mostró que el AE podría encontrar topologías que aprenden más rápidamente que la red típica totalmente conectada [40].

Un enfoque para representaciones indirectas es usar una gramática para codificar reglas que gobiernan el desarrollo de una red. En [41] se introdujo el enfoque basado en la gramática más temprana. Otro ejemplo de un sistema de desarrollo basado en gramática [42] donde cada individuo contiene las reglas para un sistema Lindenmayer (sistema-L), que son cadenas paralelas que describen las gramáticas (cada regla aplicable se usa en cada paso de la derivación). Los sistemas-L se han usado para modelar el desarrollo de organismos vivos. Para evaluar la aptitud, el sistema usa reglas del sistema-L para generar una cadena que representa la estructura de una red neuronal. Luego, la red se entrena usando backpropagation y la aptitud se determina por la combinación de la exactitud de las clasificaciones en el entrenamiento y el conjunto de prueba.

Los árboles de decisión son también métodos de clasificación populares. Debido a que la programación genética tradicionalmente usa árboles para representar las soluciones se presenta un ejemplo de Programación Genética (PG) en clasificación donde la aptitud de cada árbol de decisión está basado en la exactitud del conjunto de entrenamiento [43]. Luego se extendió la medida de aptitud para incluir términos relacionados al tamaño del árbol, y se determinó que la PG podría encontrar árboles pequeños comparables en exactitud a aquellos encontrados por C4.5 para algunos casos de prueba [44]. Otros enfoques usan la PG tradicional complementándola con un método de selección multi-objetivo que intenta minimizar simultáneamente el tamaño del árbol y los errores de la clasificación [45].

En [46] se presentó un algoritmo evolutivo interactivo que permite al usuario evaluar combinaciones de atributos que describen los datos. El objetivo del sistema es encontrar nuevas variables que pueden describir los datos concisamente y para luego utilizarse en un algoritmo de la clasificación tradicional. Cada individuo en el algoritmo usa dos árboles de PG para representar nuevas variables que son una transformación de los atributos originales. Las dos nuevas variables pueden considerarse nuevos ejes en que el conjunto de entrenamiento se proyecta y se muestra el resultado. Todos los individuos se procesan de esta manera y se presentan al usuario que decide qué proyecciones muestran algunas estructuras interesantes. Los individuos seleccionados pasan por crossover y mutación, y el ciclo se repite.

Actualmente algunos trabajos utilizan clasificadores SVM (*Support Vector Machine*) y en [19] se propone un modelo para combinar múltiples clasificadores SVM. Realizar la combinación de múltiples clasificadores es una forma natural de descubrir información útil y así mejorar la performance de los clasificadores individuales.

Uno de los desafíos de la clasificación de datos de microarray es seleccionar un número mínimo de genes relevantes que puedan maximizar la exactitud de la clasificación. Se han propuesto varios métodos de selección de genes así como sus correspondientes clasificadores. Uno de los métodos de análisis existente es un enfoque híbrido basado en un algoritmo genético y la

probabilidad de clasificación máxima (GA/MLHD). Se utiliza un algoritmo genético inteligente (IGA) que usa control de los genes y una función de fitness mejorada. Esto permite determinar el número mínimo de genes relevantes e identificar estos genes, mientras simultáneamente se maximiza la exactitud en la clasificación, la robustez de los genes seleccionados y la exactitud, especialmente para el conjunto de datos que tienen numerosas categorías y un gran número de genes de prueba en su interior [21].

La clasificación de los datos del microarray ha surgido como un tema importante en el transcurso de esta última década. Para varios métodos de selección de características y clasificadores, es muy difícil encontrar un método perfecto para clasificar los datos en el microarray en [17] se propone un algoritmo genético especializado para obtener en un tiempo razonable diversos ensambles de características y clasificadores. Los experimentos han demostrado que el método propuesto encuentra varios ensambles buenos que son superiores a los que se encuentran con clasificadores individuales.

Recientemente se ha utilizado Programación genética para la clasificación supervisada [20]. Se presenta un método supervisado de clasificación de patrones de proteoma cinético basado en programación genética que permite la extracción de patrones definidos por el usuario de una base de datos de perfiles cinéticos. El método combina la robustez del algoritmo de programación genética con la flexibilidad dada por la interacción del usuario.

### 3. Línea de Investigación

En los últimos años nuestro grupo de investigación se enfocó en el desarrollo y conocimiento de los diferentes enfoques relacionados al campo de la inteligencia computacional, en particular al de computación evolutiva [47, 48, 49, 50]. Simultáneamente, ha surgido un gran número de enfoques metaheurísticos, muchos de ellos bio-inspirados, los que a pesar de ciertas diferencias conceptuales en su diseño, comparten muchos aspectos que permiten entre otras cosas: a) aplicar conceptos que originalmente fueran diseñados para otra heurística o metaheurística con el objetivo de lograr mejoras substanciales, b) diseñar enfoques híbridos que aprovechen las ventajas relativas de cada enfoque involucrado, c) incorporar criterios de búsqueda más avanzados.

Por esta razón una de las líneas de investigación en el proyecto es la aplicabilidad de metaheurísticas como enfoque alternativo y/o complementario en tareas de minería de datos, específicamente en clasificación.

La presente propuesta tiene por objetivo desarrollar técnicas avanzadas o mejoradas de minería de datos basadas en el enfoque evolutivo. Las técnicas derivadas estarán dirigidas a tareas de clasificación supervisada y aplicadas sobre distintos problemas del mundo real. La calidad de dichas técnicas será contrastada con técnicas existentes.

### Referencias

- [1] J. I. Witten, E. Frank, Data Mining, Morgan Kaufmann Publishers, 2000.
- [2] D. Goldberg. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison Wesley, Reading, MA, 1989.

- [3] J.H. Holland. *Adaptation in natural and artificial system*. University of Michigan Press, 1975. New York, 1995.
- [4] D. Fogel. *Evolutionary Computation: Towards a NewPhilosophy of Machine Intelligence*. IEEE Press, Piscataway, NJ, 1996.
- [5] L.J. Fogel, A.J. Owens, and M.J. Walsh. *Artificial intelligence through simulated evolution*. Wiley, 1966.
- [6] I. Rechenberg. *Evolution strategy*. In J.M. Zurada, R.J. Marks II, and C. Robinson, editors, *Computational Intelligence: Imitating Life*. IEEE Press, Piscataway, NJ, 1994.
- [7] H.-P. Schwefel. *Evolution and Optimum Seeking*. Wiley, New York, 1995.
- [8] J.R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.
- [9] A.A. Freitas, *Evolutionary Computation*. J. Zytkow and W. Klogsen. (Eds.) *Handbook of Data Mining and Knowledge Discovery*. Oxford University Press, 2001.
- [10] A. A.Freitas, A survey of evolutionary algorithms for data mining and knowledge discovery. A. Ghosh and S. Tsutsui. (Eds.). *Advances in Evolutionary Computation*. Springer-Verlag, 2002.
- [11] A. Hussein, A.S. Ruhul, S.N. Charles, *DATA MINING: A heuristic approach*, Idea Group Publishing, 2002.
- [12] R. Orallo, R. Quintana, F. Ramírez, *Introducción a la minería de datos*, Pearson Education S. A., Madrid, 2004.
- [13] Sheng Chao Ding, Juan Liu , ChanLe Wu, Qing Yang, A Genetic algorithm applied to optimal gene subset selection, Congress on Evolutionary Computation CEC2004.
- [14] S.F. Smith, Searching for Protein Classification Features, Congress on Evolutionary Computation CEC2005.
- [15] L.P. Cordella, C. De Stefano, F. Fontanella , A. Marcelli, Genetic Programming for Generating Prototypes in Classification Problems; Congress on Evolutionary Computation CEC2005.
- [16] J. Gomez, A Framework for Evolving Fuzzy Rule Classifiers, Congress on Evolutionary Computation CEC2004.
- [17] Sung-Bae Cho, Chanhoo Park, Speciated GA for Optimal Ensemble Classifiers in DNA Microarray Classification, Congress on Evolutionary Computation CEC2004.
- [18] Xiujuan Chen, R. Harrison, Yan-Qing Zhang, Genetic Fuzzy Fusion of SVM Classifiers for Biomedical Data, Congress on Evolutionary Computation CEC2005.
- [19] Cuong To, Jiri Vohradsky, Classification of Proteomic Kinetic Patterns using Supervised Genetic Programming, Congress on Evolutionary Computation CEC2005.
- [20] Shinn-Ying Ho, Chong-Cheng Lee, Hung-Ming Chen, Hui-Ling Huang, Efficient Gene Selection for Classification of Microarray Data, Congress on Evolutionary Computation CEC2005.
- [21] E. Leon, O. Nasraoui, J. Gomez, Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion Detection, Congress on Evolutionary Computation CEC2004.
- [22] Weiguo Sheng, Xiaohui Liu, A Hybrid Algorithm for K-medoid Clustering of Large Data Sets, Congress on Evolutionary Computation CEC2004.
- [23] C. Spieth, F. Streichert, N. Speer, A. Zell, Clustering-based Approach to Identify Solutions for the Inference of Regulatory Networks, Congress on Evolutionary Computation CEC2005.
- [24] J.H. Holland, *Adaptation in natural and artificial systems*, Ann Arbor, MI: University of Michigan Press ,1975.
- [25] L. B Booker, D. E Goldberg, J. H. Holland, Classifier systems and genetic algorithms. *Artificial Intelligence*, 40 (1/3), 235-282, 1989.
- [26] S. F. Smith, A learning system based on genetic adaptive algorithms. *Dissertation Abstracts International*, 41 , 4582B. (University Microfilms No. 81-12638), 1980.
- [27] S. F.Smith, Flexible learning of problem solving heuristics through adaptive search. In *Proceedings of the 8th International Joint Conference on Artificial Intelligence* (pp. 422-425), 1983.
- [28] K. A De Jong, W. M Spears, D. F. Gordon, Using genetic algorithms for concept learning. *Machine Learning*, 13, 161-188, 1993.
- [29] J. Branke, *Evolutionary algorithms for neural network design and training* (Technical Report). Karlsruhe, Germany: Institute AIFB, University of Karlsruhe, 1995.
- [30] D. Whitley, Genetic algorithms and neural networks. In Winter, G., Periaux, J., Galan, M., & Cuesta, P. (Eds.), *Genetic Algorithms in Engineering and Computer Science* (Chapter 11, pp. 203-221). Chichester: John Wiley and Sons, 1995.
- [31] X.Yao, Evolving artificial neural networks. *Proceedings of the IEEE*, 87 (9), 1423-1447.
- [32] T. P. Caudell, C. P. Dolan, Parametric connectivity: Training of constrained networks using genetic algorithms. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 370-374). San Mateo, CA: Morgan Kaufmann, 1989.
- [33] D. J. Montana, L. Davis, Training feedforward neural networks using genetic algorithms. In *Proceedings 11th International Joint Conference on Artificial Intelligence* (pp. 762—767). San Mateo, CA: Morgan Kaufmann, 1989.
- [34] D.Whitley, T. Hanson, Optimizing neural networks using faster, more accurate genetic search. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 391-397). San Mateo, CA: Morgan Kaufmann, 1989.
- [35] H. Kitano, Designing neural networks using genetic algorithms with graph generation system, *Complex Systems*, 4 (4), 461-476, 1990.
- [36] A. Skinner, J. Q. Broughton, Neural networks in computational material science: training algorithms. *Modeling and Simulation in Material Science and Engineering*, 3, 371-390, 1995.
- [37] N. Kadaba, K. E. Nygard, Improving the performance of genetic algorithms in automated discovery of parameters. *Machine Learning: Proceedings of the Seventh International Conference*, 140-148, 1990.
- [38] G. F. Miller, P.M. Todd, S.U. Hegde, Designing neural networks using genetic algorithms. In Schaffer, J. D. (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 379-384). San Mateo, CA: Morgan Kaufmann, 1989.
- [39] R. Belew, J. McInerney, N. Schraudolph, Evolving networks: Using the genetic algorithm with connectionist learning (Tech. Rep. No. CS90-174). San Diego: University of California, Computer Science and Engineering Department, 1990.

- [40] D. Whitley, T. Starkweather, C. Bogart, Genetic algorithms and neural networks: Optimizing connections and connectivity. *Parallel Computing*, 14 , 347-361., 1990.
- [41] H. Kitano, Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4 (4), 461-476, 1990.
- [42] J. W. Boers, H. Kuiper, Biological metaphors and the design of modular artificial neural networks. Unpublished Master's Thesis, Leiden University, The Netherlands, 1992.
- [43] J.R. Koza, Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA: The MIT Press, 1992.
- [44] N.I. Nikolaev, V. Slavov, Inductive genetic programming with decision trees. *Intelligent Data Analysis*, 2 (1), 1998.
- [45] M.C.J. Bot, W. B. Langdon, Application of genetic programming to induction of linear classification trees. In *European Conference on Genetic Programming*, (pp. 247-258). Berlin: Springer-Verlag, 2000.
- [46] G. Venturini, M. Slimane, F. Morin, J.P. Asselin de Beauville, On using interactive genetic algorithms for knowledge discovery in databases. In Bäck, T. (Ed.), *Proceedings of the Seventh International Conference on Genetic Algorithms* (pp. 696- 703). San Francisco: Morgan Kaufmann, 1997.
- [47] M. de San Pedro, D. Pandolfi, M. Lasso, A. Villagra, Dynamic Scheduling Approaches to solve Single Machine Problem, *IASTED 2005 - International Conference on Artificial Intelligence and Soft Computing*, Benidorm Benidorm, España, Septiembre 2005.
- [48] A. Villagra, M. de San Pedro, M. Lasso, D. Pandolfi, Multirecombined Evolutionary Algorithm inspired in the Selfish Gene Theory to face the Weighted Tardiness Scheduling Problem, *Iberamia 2004 - IX Ibero-American Conference on Artificial Intelligence*, Puebla, Mexico, Noviembre 2004.
- [49] D. Pandolfi, M. Lasso, M. De San Pedro, A. Villagra, R. Gallard , Knowledge Insertion: an Efficient Approach to Reduce Search Effort in Evolutionary Scheduling *Journal of Computer Science & Technology*, Agosto 2004.
- [50] M. Lasso, D. Pandolfi, M. de San Pedro, Andrea Villagra, R. Gallard, Solving Dynamic Tardiness Problems in Single Machine Environments, *Congress on Evolutionary Computation - CEC '04*, Portland, U.S.A, Junio 2004.